

Claims

What is claimed is:

*Sub
par*

1. A method for facilitating integrity of an assembly employable by application programs during runtime, comprising the steps of:
providing an assembly with a manifest that contains a list of modules that make up the assembly; and
providing the manifest with a hash of the contents of at least one module of the list of modules.
2. The method of claim 1, the step of providing the manifest with a hash of the contents of at least one module of the list of modules comprising the step of providing the manifest with a hash of each module of the list of modules that constitutes the assembly.
3. The method of claim 1, further comprising the step of providing identity information in the manifest of the assembly.
4. The method of claim 3, the identity information comprising publisher information and version information.
5. The method of claim 1, further comprising the step of providing a hash of the contents of the assembly at the end of the assembly.
6. The method of claim 1, further comprising the step of determining if the contents of the assembly has been modified by determining an actual hash of the contents of the at least one module of the list of modules and comparing the actual hash with the hash of the contents of the at least one module of the list of modules residing in the manifest of the assembly.

Sub 1

7. The method of claim 6, further comprising the step of determining if the publisher of the assembly is trustworthy if the assembly has been modified.

8. The method of claim 7, the step of determining if the publisher of the assembly is trustworthy if the assembly has been modified comprising the step of checking version information and publisher name information residing in the manifest of the assembly.

9. The method of claim 1, further comprising the step of providing the manifest with a hash of a manifest of at least one other assembly that the assembly depends on.

10. A method for facilitating integrity of assemblies employable by application programs during runtime, comprising the steps of:

providing an assembly with a manifest that contains a list of referenced assemblies that the assembly depends on; and

providing the manifest with a hash of a manifest of at least one referenced assembly of the list of referenced assemblies.

11. The method of claim 10, the step of providing the manifest with a hash of a manifest of at least one referenced assembly of the list of referenced assemblies comprising the steps of providing the manifest with a hash of each referenced assembly of the list of referenced assemblies.

12. The method of claim 10, further comprising the step of providing identity information in the manifest of the assembly.

13. The method of claim 12, the identity information comprising publisher information and version information.

filed
for

14. The method of claim 10, further comprising the step of providing a hash of the contents of the assembly at the end of the assembly.

15. The method of claim 10, further comprising the step of determining if the contents of the at least one referenced assembly has been modified by determining an actual hash of the contents of the at least one referenced assembly of the list of referenced assemblies and comparing the actual hash with the hash of the contents of the at least one referenced assembly of the list of referenced assemblies residing in the manifest of the assembly.

16. The method of claim 15, further comprising the step of determining if the publisher of the at least one referenced assembly is trustworthy if the at least one referenced assembly has been modified.

17. The method of claim 16, the step of determining if the publisher of the at least one referenced assembly is trustworthy if the at least one referenced assembly has been modified comprising the step of checking version information and publisher name information residing in a manifest of the at least one referenced assembly.

18. A computer readable medium having at least one computer executable component employable by an application program at runtime comprising;
an assembly including a manifest that contains a list of modules that make up the assembly and a hash of the contents of at least one assembly of the list of modules.

19. The computer readable medium of claim 18, the manifest including a list of at least one referenced assembly and a hash of a manifest of the at least one referenced assembly.

20. The computer readable medium of claim 19, the manifest including identity

*Sub
ac*
information and version information.

21. The computer readable medium of claim 19, the assembly being a dynamically linked library.

22. A computer readable medium having at least one computer executable component employable by an application program at runtime comprising:
an assembly including a manifest that contains a list of at least one referenced assembly that the assembly references and a hash of the contents of a manifest of the at least one referenced assembly.

23. A system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising:
a first component adapted to provide a manifest for an assembly, the manifest having a list of modules making up the assembly; and
a second component adapted to provide the manifest with a hash of at least one module of the list of modules.

24. The system of claim 23, further comprising a third component adapted to compare the hash of the at least one module with an actual hash value of the at least one module.

25. The system of claim 24, the manifest including identity and version information and the third component adapted to determine if the assembly should be executed based on a review of the originator and version information, if the hash of the at least one module in the manifest and the actual hash value of the at least one module are different.

26. The system of claim 23, further comprising a binding component adapted to

*Sub
at*

provide binding policy information for determining a version of an assembly that an application program will run if another assembly having the same name resides on the system.

27. A system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising:
a first component adapted to provide a manifest for an assembly, the manifest having at least one referenced assembly; and
a second component adapted to provide the manifest with a hash of the at least one referenced assembly.

28. The system of claim 27, further comprising a third component adapted to compare the hash of the at least one referenced assembly in the manifest with an actual hash value of the at least one referenced assembly.

29. The system of claim 27, further comprising a binding component adapted to provide the third component with binding policy information.

30. A system for facilitating integrity of an assembly employable by application programs at runtime, the system comprising:

means for relating a manifest having a list of at least one related assembly to an assembly; and

means for providing the manifest with a hash of the at least one related assembly.

31. The system of claim 30, the at least one related assembly being a module.

32. The system of claim 30, the at least one related assembly being a referenced assembly.

of
rel

- Abstract**—The purpose of this study was to determine whether the use of a computerized decision support system (DSS) could improve the accuracy of diagnosis of patients with suspected myocardial infarction (MI). The DSS was designed to assist physicians in making a diagnosis by asking them a series of questions about the patient's history, physical examination, and laboratory test results. The DSS was used by 10 physicians who were asked to diagnose 100 patients with suspected MI. The results showed that the DSS improved the accuracy of diagnosis compared to the control group.